Balancing Reserve

Market Participants Guide
Onboarding Guide for Market Participants

1. Onboarding Steps

Single Market Platform (SMP) Registration
   1. Login to SMP
   2. Register their Asset(s)
   3. Create the Unit(s)
   4. Submit BMU ID(s)

The user guide / videos for the above can be found at Single Markets Platform | National Grid ESO

Asset/Unit Pre-qualification on Single Market Platform (SMP)
   1. Submit Pre-qualification application to National Grid Contracts Team
   2. Receive Form C from the National Grid Contracts Team
   3. Receive Unit Pre-qualified notification from the National Grid Contracts Team

The user guide / video for submitting prequalification application can be found at Single Markets Platform | National Grid ESO

Additional Onboarding Requirements for Balancing Reserve

- Market Participants need to:

   1. Provide their Admin correspondence email address to National Grid Contracts Team
   2. Provide their Auction correspondence email address to National Grid Contracts Team
   3. Provide their Company Name as found on SMP to National Grid Contracts Team
   4. IP Address(es) they will use to connect to the Balancing Reserve API – we need this information so we can whitelist the IP Address(es)

National Grid will supply necessary details on how to connect to the Balancing Reserve API.
2. Onboarding Data Flow
Bid Submission Guide for Market Participants

Option 1: Provider UI

1.1 One-time User-access setup

This user guide provides step by step instructions on how to login to the National Grid Balancing Reserve Provider Portal / UI.

Please note this is the first version of the user guide and further cosmetic enhancements are likely to occur as we go along.

Open a browser and paste the below link:

https://apps.powerapps.com/play/e/31d97878-d4ad-4d47-9bbc-ba0cffee85/a/a2ee9ae9-89d3-4adb-be3e-82cbf6bb92?tenantId=f98a6a53-25f3-4212-901c-c7787fcd3495

The above link is for the testing environment. We will update the live link once the Balancing Reserve product is live.

It will ask you to Sign in. Enter your registered email address (you should have received one from the NG Account Manager) and click on the Next button:

You will be prompted to enter the Password (you should have received one from the NG Account Manager). Enter the password and click on the Sign In button.
Select the multi factor authentication as shown below
1.2 Bid Submission - Sell Order Upload Steps

Once you have selected the multi factor authentication, it will land you on the Provider Portal / UI’s Home page as shown below:

Provider UI

Click on Upload Sell Order button to upload your Sell Order

It will take you to the Upload Sell Order page where you can find some of the useful instructions you should consider before uploading a file.
Click on Attach file and select the Sell Order file you want to submit.

Once the file is attached, click on the Run Data Validations to Proceed button.
If the Data Validations are successful, you will see a successful message saying All data validations passed.

Now click on the Upload button.
You will get the message Sell Order file Submitted successfully. You have now submitted the Sell Order file successfully.
What actions / steps the Participants should take if the Data Validation fails

Click on Upload Sell Order button to upload your Sell Order

It will take you to the Upload Sell Order page where you can find some of the useful instructions you should consider before uploading a file

Click on Attach file and select the Sell Order file you want to submit
Once the file is attached, click on the Run Data Validations to Proceed button

If Data Validations fail, you should see a message saying Data validation failed. Kindly click on Failed Data Validation button to view records that failed validations. Click on Show Data Validation Error button
It will open the Data Validation Error – Values that failed Schema Validations page and it would highlight all the errors in red. Click on each of those items to see the errors.
Make the necessary changes to the Sell Order file and upload the Sell Order file from the Home Page

Steps to ensure correct .csv encoding for Sell Orders

The National Grid API and National Grid Provider Portal / UI uses UTF-8 encoding to validate the .csv files submitted by the Market Participants. It is therefore essential that the Market Participants verify the encoding on the .csv files they are submitting and making sure the .csv files are with correct encoding format i.e. UTF-8.

The below steps will guide the Market Participants on how they can verify if the .csv file they are submitting is in the correct format:

Once the sell order has been created and saved on your internal system, right click on the .csv file you want submit and then click ‘Open With’ and then select Notepad from the available options
This will open the .csv file in notepad. On the bottom right corner, please verify if its UTF-8.

If it’s not in UTF-8 format, you need to change the encoding before you submit the bid. To change the encoding, kindly open the same file in notepad and click on save as option in the file menu at the top.
While saving the file, you will find an option to select encoding of the file. Please select UTF-8 and save the file.

Your file is now encoded in the correct format. To verify this, you can open the file on Notepad again as shown above and verify the encoding in the bottom right-hand corner shows UTF-8.
How Market Participants can cancel a Bid

In order to cancel/withdraw a Sell order (Bid), the Market Participants need to upload a new Sell Order with 0 price and 0 MW Volume offered for all the Service Windows and all the BMU Units they want to cancel/withdraw.

Note: This new Sell Order file will supersede the previous submitted file as long as the new file passes the validations. Please note that the entire file will be superseded and not just individual rows.

Option 2: Via API

Introduction
The aim is to provide a closer look at the APIs the National Grid Electricity System Operator (NG ESO) offers to enable Sell Order submission for the Balancing Reserve Auctions. This document aims to provide Market Participants with:
- An understanding of what is required to interact with the Balancing Reserve API
- An overview of the APIs published by the Balancing Reserve API

Further information
More information about Balancing Reserve, including the Balancing Reserve API, is available on the Balancing Reserve page of the NG ESO website.
To discuss opportunities offered by Balancing Reserve, please contact NG ESO via your account manager or email Commercial.Operation@nationalgrideso.com.

Overview
All Transactions with the Balancing Reserve API are synchronous, with a web service request from the Market Participant resulting in a response payload from NG ESO.

Available Balancing Reserve API Calls

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Type</th>
<th>Purpose</th>
<th>Pre-requisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate</td>
<td>Sync</td>
<td>Request an Authentication Token</td>
<td>None</td>
</tr>
<tr>
<td>Submit Sell Order</td>
<td>Sync</td>
<td>To submit a new Sell Order</td>
<td>Must have a valid authentication Token</td>
</tr>
</tbody>
</table>
## Market Participant Infrastructure Requirements

<table>
<thead>
<tr>
<th>#</th>
<th>Item</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MP Client Endpoint</td>
<td>Component</td>
<td>This is the Client Endpoint that will initiate a connection to the Balancing Reserve API.</td>
</tr>
<tr>
<td>2</td>
<td>MP Internet Gateway</td>
<td>Component</td>
<td>This is the Internet Gateway that provides connectivity between the Market Participant’s Client Endpoint and the internet. It should implement appropriate security measures. Note: NG ESO will need to know the IP address(es) of your Internet Gateway so they can whitelist them for access to the Balancing Reserve API</td>
</tr>
<tr>
<td>3</td>
<td>NG ESO Internet Gateway</td>
<td>Component</td>
<td>This is the Internet Gateway that provides connectivity between the Balancing Reserve API and the Internet. It implements security points and checks, including validating incoming flows are coming from known sources.</td>
</tr>
<tr>
<td>4</td>
<td>Balancing Reserve API Server</td>
<td>Component</td>
<td>This is the HTTPS endpoint that the Market Participant will connect to when initiating a call to the Balancing Reserve API. It will enforce that the client is using TLS1.2 or later.</td>
</tr>
<tr>
<td>5</td>
<td>HTTPS API Request</td>
<td>Flow</td>
<td>This is an instance of a connection from Market Participant to Balancing Reserve API, for example to perform an authentication request or submit a transaction.</td>
</tr>
</tbody>
</table>

### Page Break

**API: Authenticate Overview**

Market Participant to make a call to Authenticate API to get JWT Token which can then be provided to authorise access to other Balancing Reserve APIs.
<table>
<thead>
<tr>
<th>#</th>
<th>Dataflow</th>
<th>Transfer Type</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Authentication Request</td>
<td>Request</td>
<td>Authentication Credentials</td>
</tr>
<tr>
<td>0.2</td>
<td>Authentication Response</td>
<td>Response</td>
<td>One of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Authentication Error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Authentication Token</td>
</tr>
</tbody>
</table>

**Specification**

**Submission**

The submission to the Authenticate API is a POST request made up of the following parameters:

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>client_id</td>
<td>Client ID for the Balancing Reserve API; provided by NG ESO for access during registration</td>
</tr>
<tr>
<td>client_secret</td>
<td>Client Secret for the Balancing Reserve API; provided by NG ESO as part of registration</td>
</tr>
<tr>
<td>grant_type</td>
<td>Grant being requested – always &quot;password&quot;</td>
</tr>
<tr>
<td>username</td>
<td>Market Participant’s Balancing Reserve API Username; provided by NG ESO as part of registration</td>
</tr>
<tr>
<td>password</td>
<td>Market Participant’s Balancing Reserve API Password; provided by NG ESO as part of registration</td>
</tr>
<tr>
<td>Scope</td>
<td>Scope for the Balancing Reserve API; provided by NG ESO as part of registration</td>
</tr>
</tbody>
</table>

**Response**

The Authentication API will response with a JSON Payload of the following format:

```json
{
    "token_type": Always "Bearer",
    "scope": Will match the Scope provided,
    "expires_in": Time to expiry of token,
    "extExpires_in": Time to expiry of token,
    "access_token": The access token that can be used to authorise access to other Balancing Reserve API calls.
}
```

**Payload Examples**

**Submission Example**

Rather than post a sample URL, we have included a sample CURL Request as that shows the required information clearly:

```
curl --location --request POST 'https://login.microsoftonline.com/f98a6a53-25f3-4212-901c-c7787fcd3495/oauth2/v2.0/token' 
--header 'Content-Type: application/x-www-form-urlencoded'
--header 'Cookie: fpc=AsOugKFV9eNChNiNJ4sd8jj2f-MFAGAAAOC919oOAAA; stsservicecookie=estfsd;x-ms-gateway-slice=estfsd'
--data-urlencode 'client_id=my_client_id'
--data-urlencode 'client_secret=my_client_secret'
--data-urlencode 'grant_type=password'
--data-urlencode 'username=my_username'
--data-urlencode 'password=my_password'
--data-urlencode 'scope=my_scope'
```

**Response Example**
"token_type":"Bearer",
"scope":"api://57070de-35df5-4e11-9d04-47033e3048be/BR_API"
"expires_in":4918,
"ext_expires_in":4918,
"access_token":"eyJhbGciOiJIUzI1NiJ9.eyJ1bmlxZSI6IjUzIzNzYyOTQwOTQ1ODI0MDAwMzA0OG0lIiwic2Vzc2VuZGVyIjoyOTEzNSwiZXhwIjoxODc3NTI4MjQ1LCJpYXQiOjE2ODQ4NTMxMjQsImV4cCI6MTY3MjI0Nzc2NjUsImlhdCI6MTY3MjI0Nzc2NjUsImlzcyI6IjI0MDI1NjIzMSIsImV4cCI6MTY3MjI0Nzc2NjUxfQ.PZgKpDsrPS9mDMz8uIL6nMChn5QHs2lhGhqTLMz6VGk"
API: Submit Sell Order

Overview

Once an Asynchronous Transaction has been submitted and accepted, the resulting Transaction ID can be used to query the status of the transaction. If the transaction has completed, this call will return the payload that would have been sent via the resulting Notify Call.

---

### Dataflow

<table>
<thead>
<tr>
<th>#</th>
<th>Dataflow</th>
<th>Transfer Type</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Sell Order Request</td>
<td>Request</td>
<td>Sell Order JSON Array</td>
</tr>
<tr>
<td>1.2</td>
<td>Sell Order Response</td>
<td>Response</td>
<td>One of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• File Accepted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• File Rejected</td>
</tr>
</tbody>
</table>

### Specification

**Prerequisites**
The Market Participant must
- have a valid JWT token issued by the Authenticate API

**Request**
The request is a POST call with the Sell Order JSON array in the Request body, and the Authorisation JWT included in the headers. The payload of the request will be as follows:

```json
[
  [<Sell Order Block>],
  ...
  [<Sell Order Block>]
]
```

For the key-value contents contained in a <Sell Order Block>, please see [Model – Sell Order](#).

**Response**

File Accepted
If the upload passes basic format validation, the API will respond appropriately and pass onto the Balancing Reserve Auction Platform for further validation, and an internal reference generated and returned.

```json
{  "Status" : "<Generated Reference> Submitted for Final Validation"
}
```

**Note:** Confirmation of final acceptance to the auction will be received via email.

**File Rejected**

If the upload fails basic format validation, the API will respond appropriately and pass onto the Balancing Reserve Auction Platform for further validation, and an internal reference generated and returned. Confirmation of final acceptance will be received via email.

```json
{  "Status" : "File Failed Initial Validation",  "Errors" : "<Error List>"
}
```

### Payload Examples

**Request**

```json
[  {    "User Defined Order ID": "1_ABC001_PBR",    "BR Service Provider": "ABC Power Ltd",    "BMU ID": "ABC001",    "Service Type": "PBR",    "Service Start": "2023-02-24T09:00:00Z",    "Service End": "2023-02-24T09:30:00Z",    "Price": 35,    "Volume": 75,    "Linked Order": ""  },  {    "User Defined Order ID": "2_ABC001_PBR",    "BR Service Provider": "ABC Power Ltd",    "BMU ID": "ABC001",    "Service Type": "PBR",    "Service Start": "2023-02-24T09:30:00Z",    "Service End": "2023-02-24T10:00:00Z",    "Price": 41.89,    "Volume": 70,    "Linked Order": "1_ABC001_PBR"
  }
]
```

**Response – File Accepted for Further Validation**

```json
{  "Status" : "Sell_Orders_ddw_24022023_090000.csv Submitted for Final Validation"
}
```

**Response – Failed Initial Validation**

```json
{  "Status": "File Failed Initial Validation",  "Errors": "\n/9 required key [Volume] not found
/9 required key [Price] not found\n"
}
```
**Model – Sell Order**

INDEX: M/O specifies if it is Mandatory or Optional to provide value in the Payload

<table>
<thead>
<tr>
<th>M/O</th>
<th>Tag</th>
<th>Data Type</th>
<th>Example</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>User Defined Order ID</td>
<td>Text(16)</td>
<td>2_ABC001_PBR</td>
<td>Unique Order Reference for that Sell Order Line</td>
</tr>
<tr>
<td>M</td>
<td>BR Service Provider</td>
<td>Text(100)</td>
<td>ABC Power Ltd</td>
<td>Your Company Name as Registered with NG ESO</td>
</tr>
<tr>
<td>M</td>
<td>BMU ID</td>
<td>Text(10)</td>
<td>BMU ID</td>
<td>“Asset”, “Unit”</td>
</tr>
<tr>
<td>M</td>
<td>Service Type</td>
<td>Text((3)</td>
<td>PBR</td>
<td>“PBR”, “NBR”</td>
</tr>
<tr>
<td>M</td>
<td>Service Start</td>
<td>Date Time</td>
<td>2023-02-24T09:30:00Z</td>
<td>Start Date Time of Settlement Period being bid for</td>
</tr>
<tr>
<td>M</td>
<td>Service End</td>
<td>Date Time</td>
<td>2023-02-24T10:00:00Z</td>
<td>End Date Time of Settlement Period being bid for</td>
</tr>
<tr>
<td>M</td>
<td>Price</td>
<td>Number</td>
<td>41.89</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Volume</td>
<td>Integer</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>Linked Order</td>
<td>Text(16)</td>
<td>1_ABC001_PBR</td>
<td>Only include if this Line is Linked to a previous order line</td>
</tr>
</tbody>
</table>
Appendix A: General Errors
When errors are encountered in an API request, a technical error payload will be sent. The table below includes the list of error codes, descriptions and details which can be received.

<table>
<thead>
<tr>
<th>HTTPS Error Code</th>
<th>Issue</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>401 Unauthorized</td>
<td>Invalid token</td>
<td>The token has failed validation. This may be for one of the following reasons:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Token was not generated by National Grid ESO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Token has been modified so the signature fails validation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Token has expired</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Token is incomplete</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td>Invalid Grant</td>
<td>The token provided does not have a valid grant for this application. This may be because you haven’t been granted access privileges for this API. Or Username or password Incorrect</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td>Bad Request</td>
<td>If there is issue with the Payload, e.g. Mandatory fields are missing or wrong, e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Invalid datatype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Parameter has invalid length/format</td>
</tr>
<tr>
<td>404 Not Found</td>
<td>Not Found</td>
<td>Wrong URL used</td>
</tr>
<tr>
<td>500 Internal Server Error</td>
<td>Internal Server Error</td>
<td>An unknown error has occurred within the API Endpoint. If this recurs, please contact support</td>
</tr>
</tbody>
</table>

Appendix B: Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Scope of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>9th Feb 2023</td>
<td>First External Release</td>
</tr>
</tbody>
</table>